

Máquina de Turing Emulando um Codificador Binário (Nibble) com saída em 7-segmentos¹

Joélio Marinho Batista Silva
joelio.marinho@dex.ufpb.br

Joelson Nogueira de Carvalho
joelson@dex.ufpb.br

RESUMO

Nas disciplinas de computação teórica, a Máquina de Turing é sempre apresentada como um mecanismo de manipulação simbólica que é capaz de verificar procedimentos efetivos (algoritmos). Essa capacidade foi inicialmente explorada com a implementação de reconhecedores de linguagens, mas a implementação de transdutores para resolver os mais diversos problemas se deu praticamente no mesmo período. Apesar de haver alguns outros instrumentos contemporâneos que realizavam a mesma função, a Máquina de Turing tem sido historicamente reconhecida como o modelo matemático que possibilitou o desenvolvimento do computador digital. Para demonstrar a capacidade de transdução, apresenta-se neste artigo a construção de uma Máquina de Turing codificadora Binário/Gray de quatro bits com saídas codificadas em sete-segmentos, emulando circuitos clássicos da eletrônica digital.

Palavras-chave: Máquina de Turing, Codificadores, Código Gray, Código 7-segmentos

ABSTRACT

In theoretical computing disciplines, the Turing Machine is always presented as a symbolic manipulation mechanism which is capable of verifying effective procedures (algorithms). This capacity was initially explored with the implementation of language recognizers, but the implementation of transducers to solve the most diverse problems occurred practically in the very same period. Although there were some other contemporary instruments that performed such function, the Turing Machine has been historically recognized as the mathematical model that enabled the development of the digital computer. In order to demonstrate the transduction capability, this article presents the construction of a four-bit Binary/Gray coding Turing Machine with seven-segment coded outputs, emulating classic circuits of digital electronics.

¹ Trabalho de conclusão de curso, sob orientação do professor Dr. Joelson Nogueira de Carvalho submetido ao Curso de Licenciatura em Ciência da Computação do Centro de Ciências Aplicadas e Educação (CCAEE) da Universidade Federal da Paraíba, como parte dos requisitos necessários para obtenção do grau de LICENCIADO EM CIÊNCIA DA COMPUTAÇÃO.

Key words: Turing Machines, Encoders, Gray's code, 7-segments code.

1 INTRODUÇÃO.

A Máquina de Turing (MT) é uma máquina abstrata cujas principais operações limitam-se a percorrer uma fita de um lado para outro sobre esta ler, comparar e escrever símbolos. Apesar de tal restrição operacional, a MT é capaz de realizar qualquer solução algorítmica, ou seja, resolver qualquer problema que um computador digital seja capaz de resolver. Uma questão fundamental é saber como isso é possível, já que a MT não possui naturalmente capacidades algébricas ou lógicas. Esse tipo de demonstração tem sido explorado nas disciplinas teóricas da computação ao implementar MTs como transdutores, ou seja, um dispositivo que transforma algo inserido na entrada da máquina em uma saída que caracteriza a operação da máquina sobre a entrada.

Este trabalho apresenta uma máquina codificadora Binário/Gray para uma entrada de 4 bits (nibble), cujas saídas que também possuem quatro bits, são exibidas em código de sete segmentos. Essas codificações foram escolhidas sem um critério específico, mas são suficientes apenas para apresentar a MT como uma emuladora de circuitos eletrônicos que implementam funções clássicas e bem difundidas, o que já é motivo suficiente para justificar o trabalho, pois é possível verificar alguns fatores relevantes para o seu desenvolvimento; a saber:

- a. Valor educacional multidisciplinar: A demonstração da emulação de circuitos eletrônicos digitais por MTs serve como uma ferramenta educacional muito interessante, pois permite que estudantes e entusiastas de ciência da computação obtenham uma compreensão mais profunda da implementação dessas máquinas associadas ao intrincado funcionamento de circuitos eletrônicos digitais. Promovendo uma abordagem interdisciplinar para a resolução de um mesmo problema.
- b. Verificação e Depuração: Em alguns casos, a emulação de circuitos eletrônicos usando máquinas de Turing fornece um meio de verificar e depurar projetos de circuitos antes da implementação física. Por meio da simulação, os projetistas podem testar a funcionalidade e o desempenho lógico de circuitos complexos, identificando e corrigindo possíveis erros ou gargalos. Isso pode gerar uma economia de recursos, além de reduzir o risco de erros de projeto.
- c. Estabelecer uma ponte entre Teoria e Prática: Demonstrar que as máquinas de Turing podem emular circuitos eletrônicos digitais ajuda a diminuir a lacuna entre os conceitos teóricos e as aplicações práticas, ao traduzir ideias abstratas em uma solução puramente simbólica que resolve problemas que são geralmente resolvidos na prática através de

circuitos eletrônicos. Essa conexão garante que os avanços teóricos da computação encontrem caminho para solucionar diversos problemas do mundo real.

- d. Implementações de MTs emulando circuitos eletrônicos são raras de encontrar na Internet e em outras fontes como livros e revistas especializadas; nenhum artigo foi encontrado utilizando o título deste trabalho no mecanismo de busca do google.

Este trabalho tem por objetivo geral construir Máquinas de Turing capazes de realizar a mesma função de circuitos eletrônicos, especificamente criando uma MT codificadoras Binário para Gray (4 bits) onde o resultado será distribuído entre quatro MTs que traduzem cada um dos bits do nibble Gray para 7-Segmentos.

Os objetivos específicos são:

- Estudar o funcionamento dos códigos Gray e 7-Segmentos e da Máquina de Turing, para definir como a máquina pode ser usada na resolução dos problemas de codificação;
- Criar uma MTs codificadora Binário/Gray;
- Criar uma MT codificadora de bits (0 e 1) para 7-Segmentos;
- Acoplar e testar ambas as MTs codificadoras para uma determinada entrada;

O formalismo das MTs será integralmente preservado, mas serão necessárias algumas abstrações necessárias à apresentação do resultado, em termos de como se procedem as entradas e saídas desses dispositivos.

A organização deste trabalho é iniciada com a seção “Revisão Bibliográfica”, que apresenta os principais conceitos necessários ao desenvolvimento deste trabalho, seguida pela seção “Desenvolvimento”, que apresenta um diagrama de contexto que expressa a visão geral do projeto, além dos dispositivos “Máquina de Turing como transdutor”, “Codificador Binário/Gray” e “Codificador de bits binários para 7-Segmentos”. Após as definições das MTs, é apresentada a seção “Resultados Discussões” que acompanham um exemplo do funcionamento das MTs para um valor inicial dado expresso em Binário. A seção “Comentários Finais e Conclusão” encerra o trabalho.

2 REVISÃO BIBLIOGRÁFICA

2.1 Máquinas de Turing,

O conceito revolucionário de Alan Turing de uma Máquina de Turing (MT) lançou as bases para a computação moderna. Uma máquina de Turing é um modelo matemático abstrato, composto de uma fita infinitamente longa, onde símbolos podem ser lidos, escritos e sobrescritos por um cabeçote de leitura/gravação (Figura 1), controlado por um conjunto de

regras que orientam essas operações (DIVERIO & MENEZES, 2004). A MT é uma máquina puramente simbólica, tudo que ela é capaz de fazer resume-se à manipulação de símbolos, que não possuem nenhum valor numérico ou operacional; assim, o símbolo 5, por exemplo, não possui nenhuma relação aritmética a ele associada, apenas sua expressão simbólica, que o difere de outros símbolos, como a, #, 7, etc.

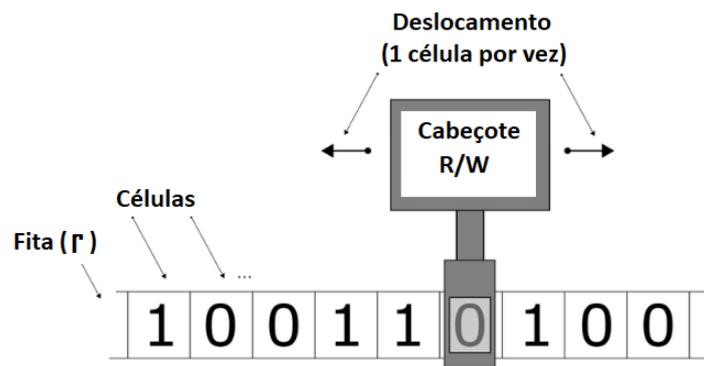


Figura 1 - Máquina de Turing – Adaptação do autor com base em SIPSER (2012).

Apesar de sua simplicidade, uma MT pode simular a computação de qualquer algoritmo, tornando-a uma poderosa construção teórica. Aliás, a MT foi originalmente construída como um modelo próprio para verificação de procedimentos efetivos, ou seja, algoritmos, tornando-se uma das ferramentas mais empregadas na Teoria da Computação para esse fim (SIPSER, 2012).

2.1.1 Máquina de Turing como transdutor

A operação fundamental de uma MT é a capacidade de transduzir uma cadeia de símbolos inicialmente colocada na fita (string de entrada), em uma outra cadeia transformada (string de saída). Esse processo de transdução é obtido por meio do uso de uma função de transição, disciplinada por estados, que define como a máquina deve operar em cada símbolo de entrada. A cada passo, a máquina lê o símbolo atual na fita de entrada, onde foi executada a transição com base no símbolo atual e escreve um novo símbolo na fita de saída (podendo esse novo símbolo ser o mesmo da entrada). Esse processo continua, até que a máquina atinja um estado de aceitação ou pare.

Pode-se definir formalmente uma **Máquina de Turing** como uma 6-tupla $\langle Q, \Sigma, \Gamma, \delta, q_0, F \rangle$, onde (DIVERIO & MENEZES, 2008):

- Q é o conjunto de estados.
- Σ é o alfabeto da entrada (não contém B).

- Γ é o alfabeto da fita, que contém Σ e eventualmente, símbolos delimitadores.
- $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \Delta$, com $\Delta = \{\{\leftarrow, -, \rightarrow\}\}$ é a função de transição da máquina de Turing.
- $q_0 \in Q$ é o estado inicial da máquina de Turing.
- $F \in Q$ é o conjunto de estados finais da máquina de Turing

As MTs transdutoras transformam uma cadeia de símbolos colocadas na fita Γ como entrada e através de passos sucessivos δ , controlados por estados Q , transforma a entrada em uma determinada saída.

2.2 O Código Binário.

O Código binário é um conjunto sequencial de números do sistema binário ou de base 2; cada um dos seus elementos é produzido a partir da soma binária de 1 unidade ao seu antecessor, assim como os naturais; a diferença é que a sua base numérica limita a representação de qualquer valor utilizando apenas dois dígitos 0 e 1, daí o nome “binário”.

Este código é fundamental para o funcionamento de computadores e outros dispositivos eletrônicos. Ele é um sistema de representação de informações que utiliza apenas dois dígitos, 0 e 1. Esses dígitos são chamados de bits. Cada bit do sistema binário representa um estado logico, onde o bit do valor 1 pode representar ligado ou verdadeiro, e o bit 0 pode representar desligado ou falso. O sistema binário é a base para internamente qualquer outro código, dado ou informação como números, texto, imagens e armazenamento de dados em disco rígido. (PESSÔA, 2023).

Tabela 1 - Conversão Decimal para Binário - Fonte: o próprio autor.

Decimal	Binário
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Um agrupamento em sequência de 8 bits é chamado de byte, e ele pode representar um número decimal na faixa de 0 a 255. Neste trabalho, será abordado um agrupamento de 4 bits, também denominado “nibble”. (IDOETA & CAPUANO, 2012).

2.3 O Código Gray.

O código binário refletido ou código Gray leva o nome do seu criador, Frank Gray, físico americano da Bell Labs que o patenteou em 1953. O código Gray é uma reordenação do sistema numérico binário de modo que dois valores sucessivos diferem em apenas um bit (dígito binário). É conhecido por código refletido, pois é formado da seguinte maneira: A partir dos dígitos 0 e 1, uma sucessão de codificações é gerado pelo posicionamento invertido dos dígitos um abaixo do outro, seguido pela inserção de zeros à esquerda da parte não invertida e uns também à esquerda, mas da parte invertida, gerando uma nova sequência. O processo se repete até que se atinja o número de bits codificados em Gray desejados (DORAN, 2007). A Figura 2 mostra o processo para 4 bits (1 nibble).

a	b	c	d	e	f	g	binário	decimal
0	0	00	00	000	000	0000	0000	0
1	1	01	01	001	001	0001	0001	1
	1	11	11	011	011	0011	0010	2
	0	10	10	010	010	0010	0011	3
			10	110	110	0110	0100	4
			11	111	111	0111	0101	5
			01	101	101	0101	0110	6
			00	100	100	0100	0111	7
					100	1100	1000	8
					101	1101	1001	9
					111	1111	1010	10
					110	1110	1011	11
					010	1010	1100	12
					011	1011	1101	13
					001	1001	1110	14
					000	1000	1111	15

a - código Gray de 1 bit
b - Reflexão
c - adiciona 0 e 1 à esquerda obtendo Gray de 2 bits
d - Reflexão
e - adiciona 0 e 1 à esquerda obtendo Gray de 3 bits
f - Reflexão
g - adiciona 0 e 1 à esquerda obtendo Gray de 4 bits

binário - Equivalente em binário
decimal - Equivalente em decimal

Figura 2 - Sucessão de reflexões e associações para gerar o código Gray.

São várias as aplicações do código Gray, dentre as quais, destacam-se:

- Este código foi inicialmente projetado para eliminar os possíveis erros causados por interruptores mecânicos não ideais. Os interruptores mecânicos que geram resultados binários sofrem erros causados por contatos múltiplos que não operam em sincronia,

de modo que os interruptores não freiam precisamente ao mesmo tempo, possibilitando leituras erradas ao alternar de um número para o próximo.

- Em matemática pode ser aplicado na resolução do problema da Torre de Hanoi.
- Já foi empregado em sistemas telegráficos, concorrendo com o código Morse.
- Minimização de circuitos booleanos (lógica digital) na rotulação dos eixos do mapa de Karnaugh.
- É utilizado em sistemas de posicionamento tanto lineares como rotativos (encoder). Inicialmente era eletromecânico e atualmente utiliza sensores óticos, de efeito hall ou usa outras tecnologias eletrônicas.
- O código Gray é utilizado na transmissão de sinais digitais, pois minimiza a ocorrência de erros.
- em conversores digitais para analógico e conversores analógicos para digitais
- O código Gray é preferível ao código binário direto em dispositivos de medição de ângulo. O uso do código Gray quase elimina a possibilidade de leitura incorreta do ângulo, o que é provável se o ângulo for representado em código binário sequencial. A propriedade cíclica do código Gray é uma vantagem nesta aplicação.
- O código Gray é usado para rotular os eixos dos mapas de Karnaugh, uma técnica gráfica usada para a minimização de expressões booleanas.
- O uso de códigos Gray para endereçar a memória do programa em computadores minimiza o consumo de energia. Isso se deve a menos linhas de endereço que mudam de estado com avanços no contador do programa.
- Os códigos Gray também são muito úteis em algoritmos genéticos, pois as mutações no código permitem principalmente alterações incrementais. No entanto, ocasionalmente, uma alteração de um bit pode resultar em um grande salto, levando a novas propriedades.

Não faz parte dos objetivos deste trabalho detalhar como cada uma das aplicações do código Gray funciona e quão relevantes são essas aplicações, mas todas essas importantes aplicações podem ser detalhadas recorrendo-se às referências indicadas (DORAN, 2007), (MAINI, 2007).

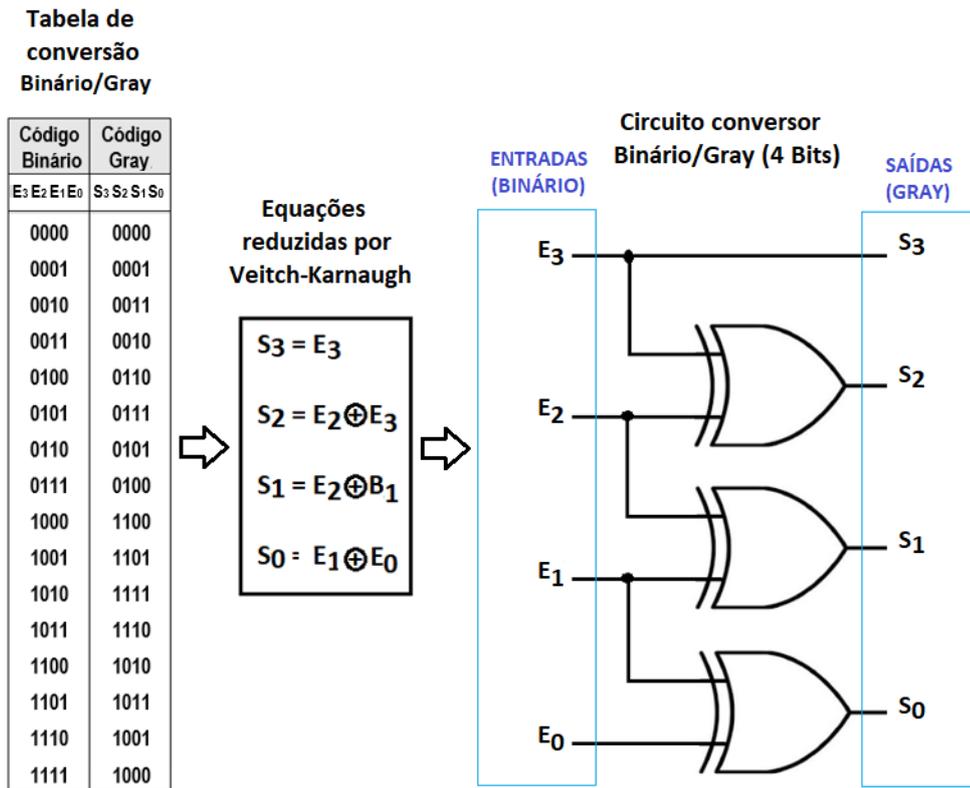


Figura 3- Codificador Binário/Gray (4 bits). Fonte: Notas de aulas do prof. Joelson Carvalho-DCX/UFPB.

A Figura 3 apresenta um circuito codificador Binário/Gray (4 Bits). A codificação cheia (16 símbolos), como a expressa a tabela inserida nesta figura, foi escolhida para este trabalho, pois mapeiam todas as possíveis combinações de valores de entrada (nibble) para o código Gray. As expressões lógicas Booleanas que representam as conversões mostradas na tabela, estão reduzidas ao máximo pelo diagrama de redução de Veitch-Karnaugh, e o circuito codificador é construído com o auxílio de três portas lógicas XOR (NOGUEIRA, 2011).

2.4 O Código 7-Segmentos.

O código de 7 segmentos é uma forma de representar números e algumas letras usando sete segmentos individuais organizados em um padrão de exibição comum. Cada segmento é ativado ou desativado individualmente, para exibir um caractere específico. Esses segmentos são organizados pelas letras A, B, C, D, E, F e G (TOCCI et al., 2007). A ativação de cada segmento se dá pela associação de valores de tensão, representados pelos dígitos binários 0 e 1, às letras que representam o segmento, como mostra a Figura 4. Assim, para exibir o número "0", por exemplo, todos os segmentos, exceto o segmento G, são ativados (1). Para representar o "1", apenas os segmentos B e C são ativados.

Displays de 7-segmentos podem ser facilmente interligados a circuitos que exprimem uma saída alfanumérica, tornando-se uma tecnologia de exibição versátil e amplamente utilizada (PUHLMANN, 2023). Esses displays são comumente encontrados em dispositivos eletrônicos que utilizam o 7-segmentos, a exemplo: os relógios digitais, as calculadoras, instrumentos de medição, enfim, qualquer tipo de “*digital gauge*”.

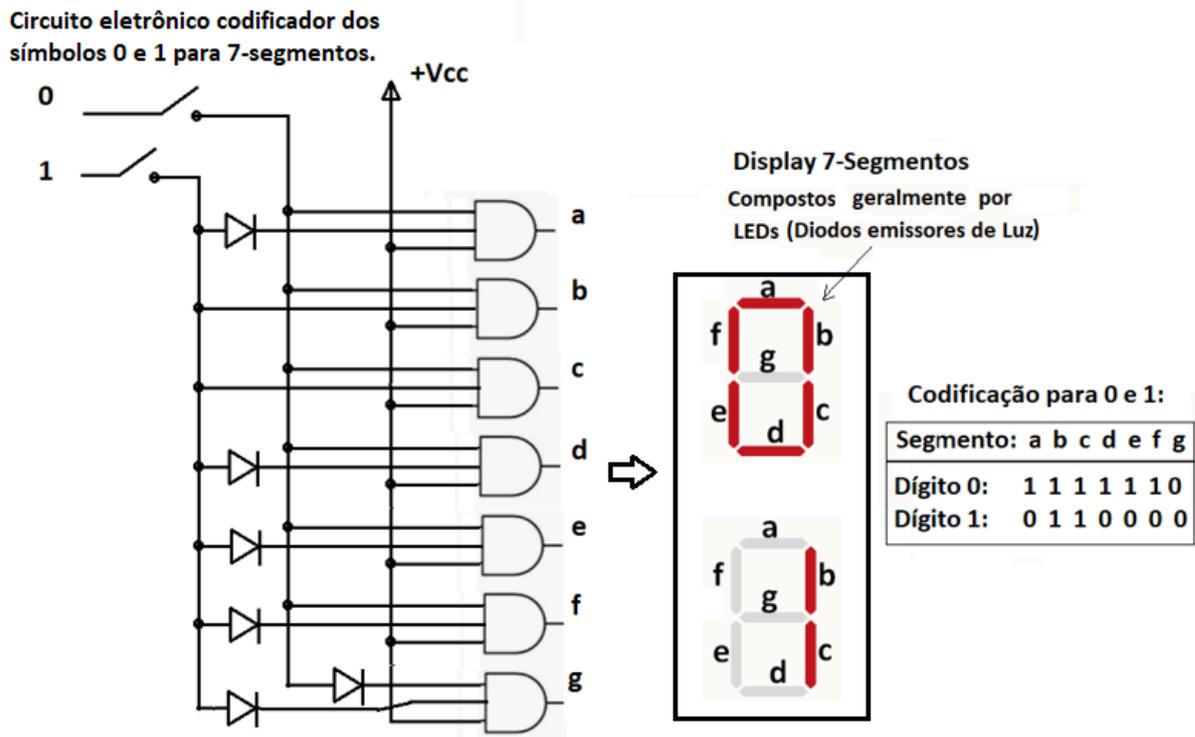


Figura 4 - Codificador Binário (0 e 1)/7-Segmentos. Fonte: Abstração do autor.

A Figura 4 apresenta um codificador dos dígitos binários 0 e 1 para 7-Segmentos; nela, é possível observar a disposição dos 7 segmentos que formam um display capaz de representar uma certa quantidade de símbolos distintos e uma tabela de codificação para a representação dos dígitos binários neste dispositivo. O circuito apresenta a montagem eletrônica do codificador utilizando portas lógicas AND e inversores. Pode ainda ser observada que as entradas do circuito são chaves que habilitam os símbolos 0 ou 1; as chaves são acionadas alternadamente, nunca simultaneamente, e a saída está designada por a,b,c,d,e,f e g, que são os caracteres associados às entradas do display de 7 segmentos.

3 DESENVOLVIMENTO

3.1 O Diagrama de Contexto

O diagrama de contexto deste trabalho é apresentado na Figura 5; a ideia proposta para essa implementação é que haverá uma MT para converter um conjunto de 4 bits Binário, inseridos

numa entrada conectada à MT que o codifica para GRAY, e tem sua saída individualizada e conectada à quatro MTs idênticas, que convertem cada bit do nibble codificado em GRAY num padrão de 7-segmentos. As saídas de cada uma das quatro MT-7SEG são então diretamente conectadas aos pinos de entrada de cada um dos 4 displays de 7-Segmentos.

Como a MT codificadora terá o papel de emular circuitos digitais, dispositivos que seriam componentes dos circuitos reais são aqui abstraídas em um modelo virtual, com o objetivo de aproximar ao máximo esse modelo da sua versão eletrônica; exemplo disso são as representações das chaves que definem o *Nibble* de entrada em Binário no codificador MT-Binário/GRAY, as conexões físicas entre os módulos MT-Binário/GRAY e as MT-7SEG e cada módulo de display digital de 7 segmentos. Apesar disso, esse modelo é totalmente passível de ser implementado fisicamente, com o uso de uma plataforma como o ARDUINO (ZELENOVSKY, 2019), por exemplo. As diversas MT poderiam ser programadas nessa plataforma e as entradas e saídas estariam dispostas no próprio dispositivo, possibilitando uma grande aproximação do modelo aqui proposto.

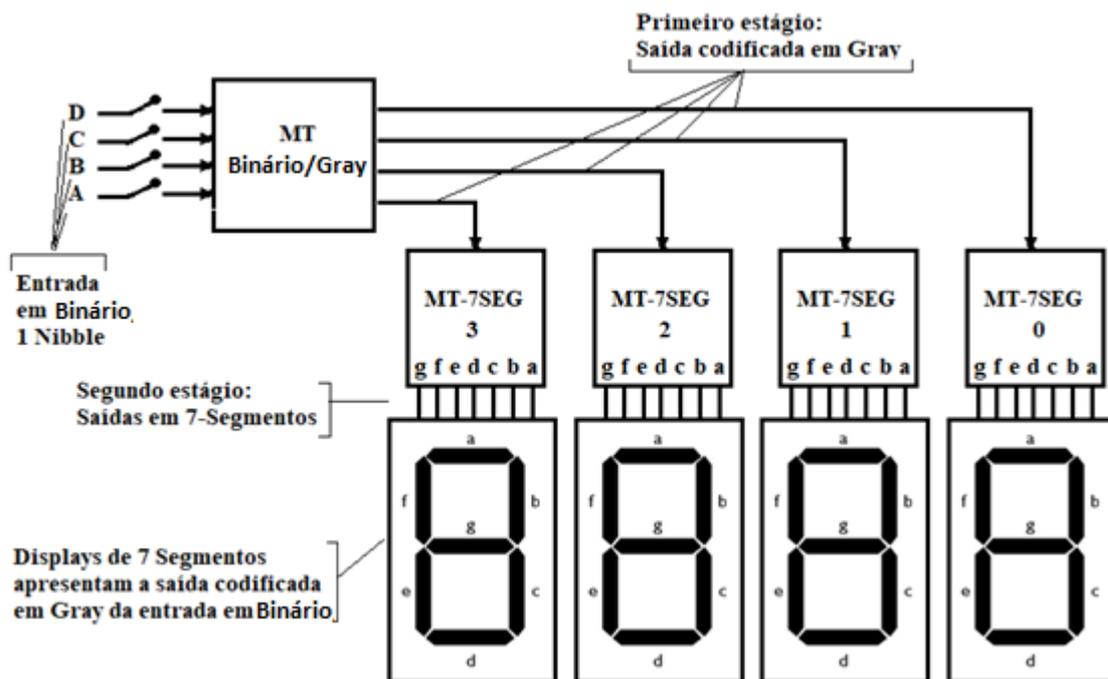


Figura 5 - Diagrama funcional do projeto. Fonte: Adaptado das notas de aula do prof. Joelson Nogueira –DCX/CCAUE/UFPB.

3.2 Máquina de Turing como tradutora de códigos Binário/GRAY

A MT codificadora Binário/Gray receberá como entrada um número binário de 4 bits e deverá emitir como resposta, o código Gray equivalente; esta codificação Binário/Gray está definida na tabela 2.

Tabela 2- Conversão Binário (4 Bits)/Gray. Fonte: Próprio do autor

Código Binário	Código Gray
0000	0000
0001	0001
0010	0011
0011	0010
0100	0110
0101	0111
0110	0101
0111	0100
1000	1100
1001	1101
1010	1111
1011	1110
1100	1010
1101	1011
1110	1001
1111	1000

3.2.1 Definição da Interface de I/O:

Inicialmente, é necessário definir a configuração da Máquina de Turing BINÁRIO-GRAY, definindo o local onde será realizada a entrada BINÁRIO e também a saída codificada GRAY. Uma configuração simples seria como mostra a Figura 6, onde Γ_i representa a fita com a entrada inserida e Γ_f representa a fita com a saída impressa como resposta à entrada.

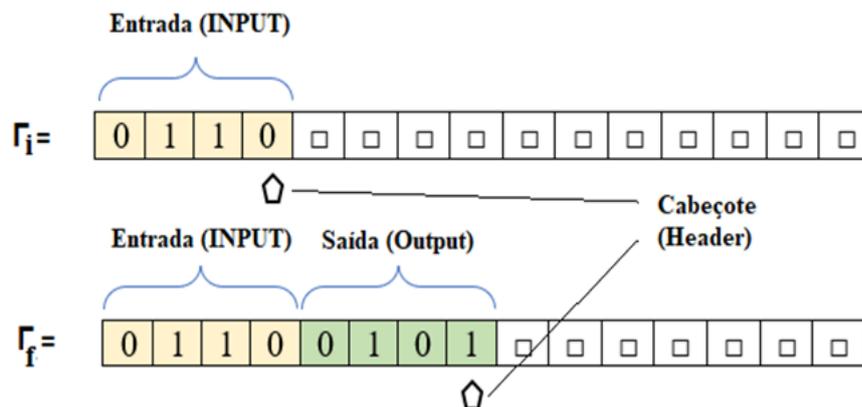


Figura 6 - - Configuração de um MT BINÁRIO-GRAY. Fonte: Abstração do autor.

3.2.2 Implementação da Máquina de Turing Codificadora BINÁRIO/GRAY:

A Máquina de Turing pôde então ser definida da seguinte forma:

- $\Sigma = \Gamma = \{0,1\}$

- $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \Delta$. onde $\Delta = \{\leftarrow, \rightarrow, -\}$; a tabela 3 apresenta as funções de transição:

Tabela 3 - Tabela da função programa da MT codificador BINÁRIO/Gray. Fonte: Próprio do autor.

Q	Γ	\rightarrow	Q	Γ	Δ	Msg Acc
q0	0		q1	0	\rightarrow	0
q0	1		q2	1	\rightarrow	1
q1	0		q3	0	\rightarrow	00
q1	1		q4	1	\rightarrow	01
q2	0		q5	0	\rightarrow	10
q2	1		q6	1	\rightarrow	11
q3	0		q7	0	\rightarrow	000
q3	1		q8	1	\rightarrow	001
q4	0		q9	0	\rightarrow	010
q4	1		q10	1	\rightarrow	011
q5	0		q11	0	\rightarrow	100
q5	1		q12	1	\rightarrow	101
q6	0		q13	0	\rightarrow	110
q6	1		q14	1	\rightarrow	111
q7	0		q15	0	\rightarrow	0000
q7	1		q16	1	\rightarrow	0001
q8	0		q17	0	\rightarrow	0010
q8	1		q18	1	\rightarrow	0011
q9	0		q19	0	\rightarrow	0100
q9	1		q20	1	\rightarrow	0101
q10	0		q21	0	\rightarrow	0110
q10	1		q22	1	\rightarrow	0111
q11	0		q23	0	\rightarrow	1000
q11	1		q24	1	\rightarrow	1001
q12	0		q25	0	\rightarrow	1010
q12	1		q26	1	\rightarrow	1011
q13	0		q27	0	\rightarrow	1100
q13	1		q28	1	\rightarrow	1101
q14	0		q29	0	\rightarrow	1110
q14	1		q30	1	\rightarrow	1111
q15	\square		q15a	0	\rightarrow	Imprime 0000
q15a	\square		q15b	0	\rightarrow	
q15b	\square		q15c	0	\rightarrow	
q15c	\square		qf	0	-	
q16	\square		q16a	0	\rightarrow	Imprime 0001
q16a	\square		q16b	0	\rightarrow	
q16b	\square		q16c	0	\rightarrow	
q16c	\square		qf	1	-	
q17	\square		q17a	0	\rightarrow	Imprime 0011
q17a	\square		q17b	0	\rightarrow	
q17b	\square		q17c	1	\rightarrow	
q17c	\square		qf	1	-	
q18	\square		q18a	0	\rightarrow	Imprime 0010
q18a	\square		q18b	0	\rightarrow	
q18b	\square		q18c	1	\rightarrow	
q18c	\square		qf	0	-	

q19	\square		q19a	0	\rightarrow	Imprime 0110
q19a	\square		q19b	1	\rightarrow	
q19b	\square		q19c	1	\rightarrow	
q19c	\square		qf	0	-	Imprime 0111
q20	\square		q20a	0	\rightarrow	
q20a	\square		q20b	1	\rightarrow	
q20b	\square		q20c	1	\rightarrow	
q20c	\square		qf	1	-	Imprime 0101
q21	\square		q21a	0	\rightarrow	
q21a	\square		q21b	1	\rightarrow	
q21b	\square		q21c	0	\rightarrow	
q21c	\square		qf	1	-	Imprime 0100
q22	\square		q22a	0	\rightarrow	
q22a	\square		q22b	1	\rightarrow	
q22b	\square		q22c	0	\rightarrow	
q22c	\square		qf	0	-	Imprime 1100
q23	\square		q23a	1	\rightarrow	
q23a	\square		q23b	1	\rightarrow	
q23b	\square		q23c	0	\rightarrow	
q23c	\square		qf	0	-	Imprime 1101
q24	\square		q24a	1	\rightarrow	
q24a	\square		q24b	1	\rightarrow	
q24b	\square		q24c	0	\rightarrow	
q24c	\square		qf	1	-	Imprime 1111
q25	\square		q25a	1	\rightarrow	
q25a	\square		q25b	1	\rightarrow	
q25b	\square		q25c	1	\rightarrow	
q25c	\square		qf	1	-	Imprime 1110
q26	\square		q26a	1	\rightarrow	
q26a	\square		q26b	1	\rightarrow	
q26b	\square		q26c	1	\rightarrow	
q26c	\square		qf	0	-	Imprime 1010
q27	\square		q27a	1	\rightarrow	
q27a	\square		q27b	0	\rightarrow	
q27b	\square		q27c	1	\rightarrow	
q27c	\square		qf	0	-	Imprime 1011
q28	\square		q28a	1	\rightarrow	
q28a	\square		q28b	0	\rightarrow	
q28b	\square		q28c	1	\rightarrow	
q28c	\square		qf	1	-	Imprime 1001
q29	\square		q29a	1	\rightarrow	
q29a	\square		q29b	0	\rightarrow	
q29b	\square		q29c	0	\rightarrow	
q29c	\square		qf	1	-	Imprime 1000
q30	\square		q30a	1	\rightarrow	
q30a	\square		q30b	0	\rightarrow	
q30b	\square		q30c	0	\rightarrow	
q30c	\square		qf	0	-	

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12}, q_{13}, q_{14}, q_{15}, q_{15a}, q_{15b}, q_{15c}, q_{16}, q_{16a}, q_{16b}, q_{16c}, q_{17}, q_{17a}, q_{17b}, q_{17c}, q_{18}, q_{18a}, q_{18b}, q_{18c}, q_{19}, q_{19a}, q_{19b}, q_{19c}, q_{20}, q_{20a}, q_{20b}, q_{20c}, q_{21}, q_{21a}, q_{21b}, q_{21c}, q_{22}, q_{22a}, q_{22b}, q_{22c}, q_{23}, q_{23a}, q_{23b}, q_{23c}, q_{24}, q_{24a}, q_{24b}, q_{24c}, q_{25}, q_{25a}, q_{25b}, q_{25c}, q_{26}, q_{26a}, q_{26b}, q_{26c}, q_{27}, q_{27a}, q_{27b}, q_{27c}, q_{28}, q_{28a}, q_{28b}, q_{28c}, q_{29}, q_{29a}, q_{29b}, q_{29c}, q_{30}, q_{30a}, q_{30b}, q_{30c}\}.$
- Estados Finais = $\{q_f\}$

3.2.3 Definição do algoritmo de codificação:

O Algoritmo a ser implementado pela MT é bastante simples; a partir da entrada da palavra a ser codificada, o cabeçote de I/O encontra-se no início da fita Γ . A Máquina de Turing deve iniciar reconhecendo cada caractere 0 ou 1 em cada célula, e em função disso assume diferentes estados para cada símbolo lido e desloca-se sempre para a direita, em busca de reconhecer o próximo caractere; repetindo-se por 4 vezes (tamanho do Nibble). Como para cada célula, o cabeçote pode reconhecer caracteres binários, no final da leitura, a Máquina de Turing deverá reconhecer $2^4=16$ diferentes sequências de 4 bits.

A parte final é a impressão do código GRAY; como já foram identificadas as 16 possíveis combinações de entrada, a MT já associa cada uma delas a um único estado, cada um desses 16 estados iniciará a impressão do seu respectivo código, criando 4 estados adicionais para imprimir o resultado correspondente em células vazias da fita da MT codificadora.

3.3 Máquina de Turing como codificador Binários (0 e 1)/7-SEGMENTOS

3.3.1 Definição da Interface de I/O:

A configuração da fita Γ da Máquina de Turing 7-SEGMENTOS é mostrada na Figura 7. Nessa configuração, apenas a primeira célula é preenchida na entrada, já que esse decodificador se limitará a codificar apenas dois dígitos, o 0 e o 1. A saída será preenchida após o funcionamento da MT, inserindo nas próximas 7 células, o código de sete segmentos correspondente ao valor inserido na entrada.

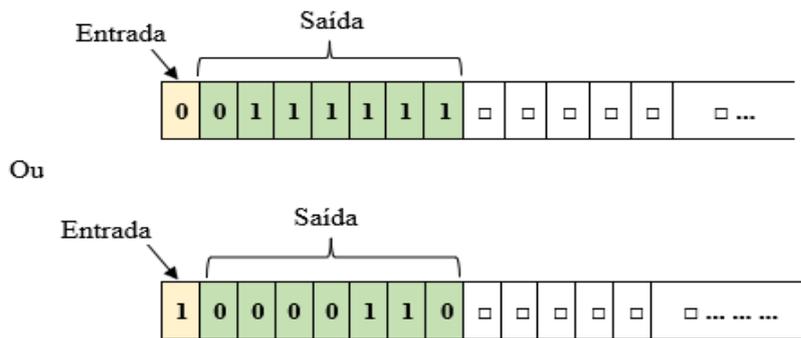


Figura 7 - Configuração de Entrada da MT-7SEG; Fonte: Abstração do autor

3.3.2 Implementação da Máquina de Turing codificadora GRAY/7-SEGMENTOS

Tabela 4 - Codificação BINÁRIO (0 e 1)/7-SEGMENTOS. Fonte: Próprio autor.

		7-SEGMENTOS						
		g	f	e	d	c	b	a
BINÁRIO	0	0	0	0	0	0	0	0
	0	0	1	1	1	1	1	1
	1	0	0	0	0	1	1	0

Como foi mostrado neste trabalho, um decodificador GRAY para 7-SEGMENTOS necessita decodificar apenas dois caracteres para serem exibidos em um display: 0 e 1. A tabela 4 apresenta a codificação do código binário para o código 7-Segmentos; pode-se observar que a codificação para os dígitos 0 e 1, que são usados para representar os bits do código GRAY de 4 bits, é obtida a partir de duas linhas dessa tabela (hachuradas em verde).

3.3.3 Implementação da MT Codificadora GRAY/7SEG:

A Máquina de Turing pôde então ser definida da seguinte forma:

- $\Sigma = \{0,1\}$
- $Q = \{q_0, q_1, q_{1a}, q_{1b}, q_{1c}, q_{1d}, q_{1e}, q_{1f}, q_2, q_{2a}, q_{2b}, q_{2c}, q_{2d}, q_{2e}, q_{2f}\}$
- $F = \Gamma = [q_f]$.
- $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \Delta$. onde $\Delta = \{\leftarrow, \rightarrow, -\}$; a tabela 5 apresenta as funções de transição:

Tabela 5 - da função programa da MT codificador Binário(0 e 1)/7SEG. Fonte: Próprio autor.

Q	Γ	\rightarrow	Q	Γ	Δ	Msg
q0	0		q1	0	\rightarrow	0
q0	1		q2	1	\rightarrow	1
q1	\square		q1a	0	\rightarrow	Imprime 0111111
q1a	\square		q1b	1	\rightarrow	
q1b	\square		q1c	1	\rightarrow	
q1c	\square		q1d	1	\rightarrow	
q1d	\square		q1e	1	\rightarrow	
q1e	\square		q1f	1	\rightarrow	
q1f	\square		qf	1	-	
q2	\square		q2a	0	\rightarrow	Imprime 0000110
q2a	\square		q2b	0	\rightarrow	
q2b	\square		q2c	0	\rightarrow	
q2c	\square		q2d	0	\rightarrow	
q2d	\square		q2e	1	\rightarrow	
q2e	\square		q2f	1	\rightarrow	
q2f	\square		qf	0	-	

3.3.4 Definição do algoritmo de codificação:

O Algoritmo a ser implementado pela MT é bastante simples; a partir da entrada da palavra a ser codificada, o cabeçote de I/O encontra-se no início da fita Γ . A MT deve iniciar reconhecendo cada caractere 0 ou 1 em cada célula, e em função disso, assumir os diferentes estados para cada símbolo lido definidos na função programa δ , deslocando-se sempre para a direita em busca de reconhecer o próximo caractere.

A parte final é a impressão do código GRAY/7-Segmentos; como já foram identificadas as possíveis combinações de entrada, a Máquina de Turing associa cada uma delas a um único estado, cada um desses 2 estados iniciará a impressão do seu respectivo código, criando 7 estados adicionais para imprimir o resultado correspondente.

4 RESULTADOS E DISCUSSÕES

Para um melhor entendimento do funcionamento da MTs BINARIA/Gray e Gray/7-segmentos, apresenta-se a execução para a entrada de um número binário 0111.

A MT binaria/Gray inicia no estado q_0 lendo o valor 0 na fita, reescreve o valor 0 e passara para o estado q_1 . No estado q_1 , a MT lê o valor 1 reescrevendo o mesmo e passando para o estado q_4 . Em q_4 , a MT lê o valor 1 e reescreve o valor 1, passando para o estado q_{10} .

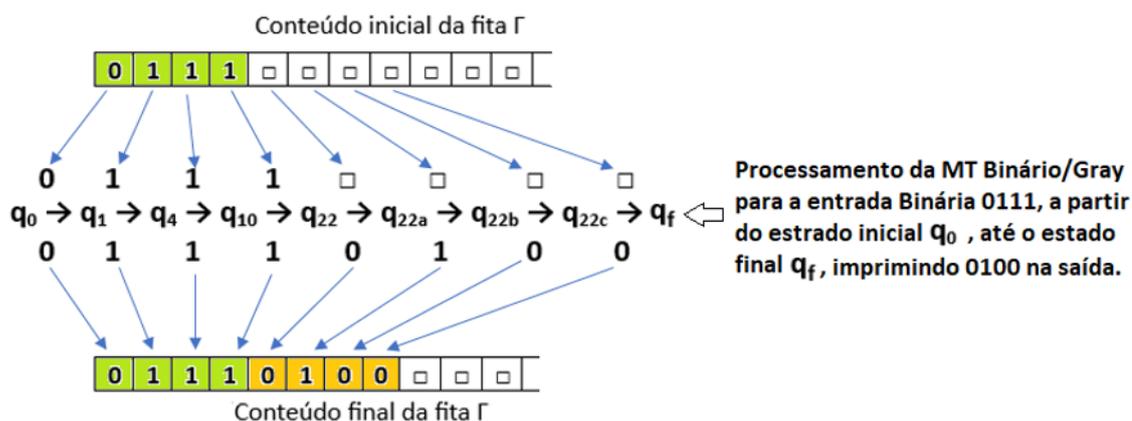


Figura 8 - execução da MT codificadora binaria/Gray, para a entrada 0111; Fonte: Próprio autor

No estado q_{10} , a MT lê o valor 1 e reescreve na fita o mesmo valor, passando para o estado q_{22} . Em q_{22} será iniciada a impressão do nibble Gray, não lendo mais nada na fita. O estado q_{22a} imprime 0, q_{22b} imprime 1, q_{22c} imprime 0 e finaliza em q_f imprimindo 0 (Figura 8).

A partir do resultado da MT binaria/Gray, quatro réplicas da MTs Gray/7-segmentos, com entradas provenientes de cada bit do nibble Gray para uma MT Gray/7-segmentos, são ativadas; são elas: MT Gray/7seg0, MT Gray/7seg1, MT Gray/7seg2 e MT Gray/7seg3

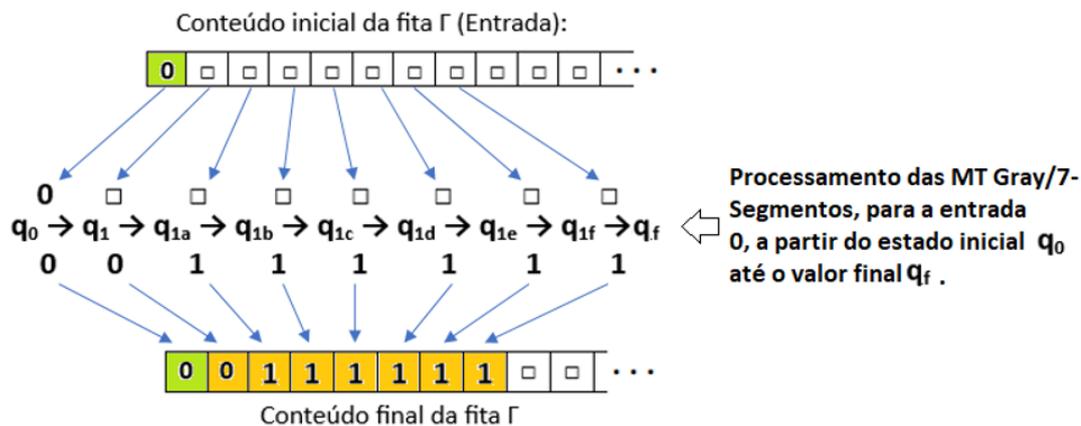


Figura 9 - execução da MT codificadora Gray/7-Segmentos, para a entrada 0; Fonte: Próprio autor

Quando o estado da MT q_0 estiver lendo 0, mudara para o estado q_1 onde os estados q_{1a} , q_{1b} , q_{1c} , q_{1d} , q_{1e} , q_{1f} e q_f imprimi os bits que indicam os 7-segmentos do display para valor 0 (Figura 9).

Quando o estado da MT q_0 estiver lendo 1, mudara para o estado q_2 onde os estados q_{2a} , q_{2b} , q_{2c} , q_{2d} , q_{2e} , q_{2f} e q_f imprimi os bits que indicam os 7-segmentos do display para valor 1 (Figura 10).

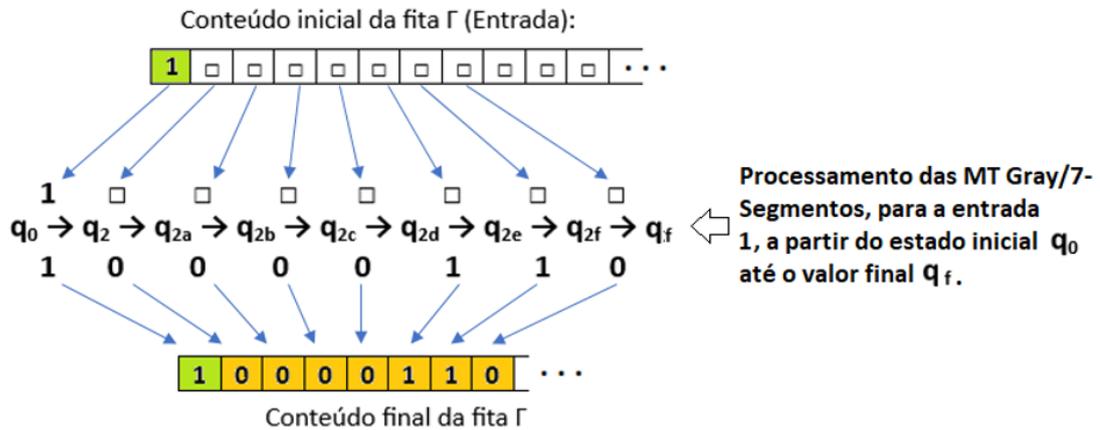


Figura 10 - execução da MT codificadora Gray/7-Segmentos, para a entrada 1; Fonte: Próprio autor

Assim, a MT termina alcançando o resultado que pode ser visto na Figura 11, para a entrada BINÁRIO 0111, o codificador BINÁRIO/Gray que apresenta o resultado 0100.

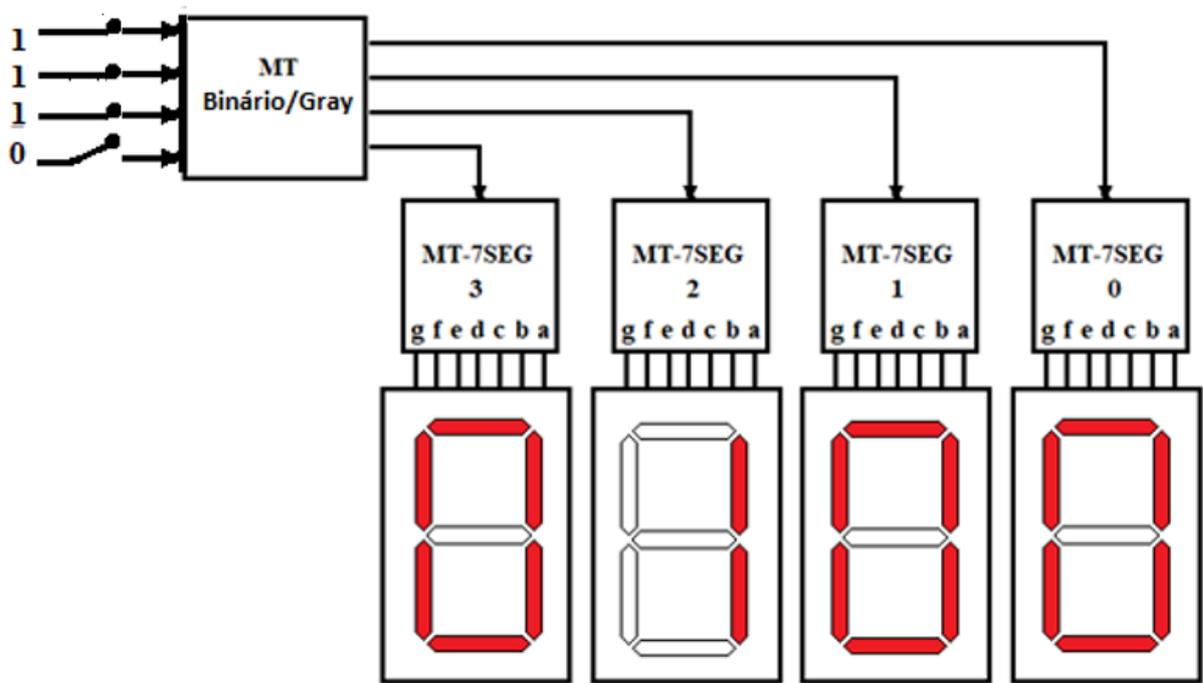


Figura 11 - Resultado da inserção do valor 0111(BINÁRIO) na entrada, com a saída exibida nos displays de 7 segmentos. Fonte: Adaptação da Figura 1 pelo autor.

Deve ser observado, que o código BINÁRIO de entrada possui o dígito mais significativo (0) na última chave. A saída mais significativa desse codificador está ligada à MT-7SEGMENTOS 3, a menos significativa, está ligada à MT-7SEGMENTOS 0.

5 CONSIDERAÇÕES FINAIS

Este trabalho apresenta a construção de uma MT capaz de converter código binário em código Gray, e uma MT que codifica cada bit da saída Gray em 7 segmentos. Demonstrando como é possível criar um algoritmo capaz de emular codificadores utilizando a capacidade das MT de processar símbolos e realizar transições de estado de acordo com regras pré-definidas.

O trabalho apresenta a definições e alguns fundamentos da MT, bem como informações sobre os códigos aqui mencionados – Gray e 7-Segmentos.

Máquinas de Turing capazes de implementar as codificações Binário/Gray e bits 0 e 1 para 7-Segmentos foram construídas e testadas, os resultados foram apresentados passo a passo e a entrada e saída representadas em um diagrama de contexto que espelha a interligação dos codificadores e as conexões entre as entradas e saídas e as MTs codificadoras. Os codificadores implementados apresentaram respostas corretas para as entradas apresentadas.

A principal dificuldade encontrada neste trabalho foi a falta de exemplos similares na literatura.

Trabalhos posteriores serão propostos, como a implementação de outros codificadores e até mesmo uma implementação física dos codificadores apresentados neste trabalho em dispositivos de estados físicos baseados na tecnologia Arduino (ZELENOVSK, 2019).

6 REFERÊNCIAS BIBLIOGRÁFICAS

DIVERIO, T. A.; MENEZES, P.F.B.: **Teoria da Computação – Máquinas Universais e Computabilidade**. Porto Alegre: Bookman, 2008.

DORAN, J.W.: **The Gray Code** - Journal of Universal Computer Science, vol. 13, no. 11 2007. Disponível na URL: https://www.jucs.org/jucs_13_11/the_gray_code/jucs_13_11_1573_1597_doran.pdf; acessado em 15/04/2023.

IDOETA, I. V., CAPUANO, F. G. Elementos de eletrônica digital, São Paulo: Érica, 2012.

MAINI, A. K.: **Digital electronics: principles, devices and applications**. 1ª Ed. John Wiley & Sons, 2007.

NOGUEIRA, J.: **Eletrônica Digital Básica**, EDUFBA, 2011. disponível na URL https://repositorio.ufba.br/bitstream/ri/13988/1/_Eletronica.pdf, acessado em 33 de maio de 2023.

PESSÔA, C.: **Entenda o sistema de Código Binário**. Disponível na URL:<https://www.alura.com.br/artigos/sistema-codigo-binario>. visitado em 20 de maio de 2023.

PUHLMANN, H. F. W.: **Sistemas Operacionais de Tempo Real – Displays de 7 segmentos**. ResearchGate, 2014. Disponível em: https://www.researchgate.net/publication/277954139_Sistemas_Operacionais_de_Tempo_Real_-_Displays_de_7_segmentos. Acesso em: 20 mar. 2023.

SIPSER, Michael: **Introdução à Teoria da Computação**, 2ª Edição. Cengage Learning, 2012.

TOCCI, R.J, WIDMER, N.S. e MOSS, G.L.: **Sistemas Digitais – Princípios e Aplicações**, 10ª Edição. Pearson Prentice Hall. 2007.

ZELENOVSKY, R.: **Arduino: Guia Avançado Para Projetos**. Editora Interciência, 2019.